

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	David E. Lowell, et al.	§	Art Unit:	2191
		§		
Serial No.:	10/676,557	§	Confirmation No.:	7663
		§		
Filed:	October 1, 2003	§	Examiner:	Qing Chen
		§		
For:	Interposing a Virtual	§	Atty. Dkt. No.:	200208633-1
	Machine Monitor and	§		(HPC.0522US)
	Devirtualizing Computer	§		
	Hardware	§		

**Mail Stop Amendment**

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

SUMMARY OF TELEPHONIC INTERVIEW

Sir:

On or around October 11, 2010, Examiner Qing Chen called the undersigned to propose amendments to allow the case. On or around November 19, 2010, the undersigned sent an unofficial amendment with amendments of claims modified from the proposal by the Examiner. After further discussions regarding further minor changes to the claims, agreement was reached that the proposed amendments would be entered by Examiner's Amendment to allow the case.

No references or exhibits were discussed.

Respectfully submitted,

Date: November 22, 2010

/Dan C. Hu/

Dan C. Hu  
Registration No. 40,025  
TROP, PRUNER & HU, P.C.  
1616 South Voss Road, Suite 750  
Houston, TX 77057-2631  
Telephone: (713) 468-8880  
Facsimile: (713) 468-8883

1. (Currently Amended) A method of using a virtual machine monitor and an operating system on computer hardware in a computer, the method comprising:

booting the operating system on the computer hardware such that the operating system has direct control of at least a portion of the computer hardware including a central processing unit (CPU), a physical memory, and an input/output (I/O) device;

interposing the virtual machine monitor between the computer hardware and the operating system at runtime, wherein runtime includes a period of execution in the computer after booting and before shutdown of the computer, wherein the interposing occurs after booting of the computer, and wherein interposing the virtual machine monitor gives the virtual machine monitor direct control of the at least a portion of the computer hardware such that the operating system no longer has direct control of the at least a portion of the computer hardware; and

~~booting the operating system on the computer hardware before interposing the virtual machine monitor at runtime~~

devirtualizing the at least a portion of the computer hardware at runtime after the virtual machine monitor has been interposed at runtime, wherein devirtualizing the at least a portion of the computer hardware comprises stopping the virtual machine monitor such that the operating system resumes having direct control of the at least a portion of the computer hardware.

2.-4. (Cancelled)

5. (Currently Amended) The method of claim 1, ~~wherein the computer hardware includes a CPU; and~~ wherein the virtual machine monitor is interposed on the CPU.

6. (Currently Amended) The method of claim 5, ~~wherein the computer hardware further includes memory, and~~ the virtual machine monitor and the operating system each include CPU interrupt handlers; and wherein interposing the virtual machine monitor on the CPU includes:

- causing privileged instructions to trap to the virtual machine monitor, and
- redirecting interrupts to the corresponding virtual machine monitor CPU interrupt handlers instead of to the operating system CPU interrupt handlers.

7. (Original) The method of claim 6, wherein the privileged instructions are caused to trap to the virtual machine monitor by causing the operating system to run at a reduced privilege level; and wherein interposing the virtual machine monitor on the CPU further includes returning control to the operating system at the reduced privilege level.

8. (Previously Presented) The method of claim 6, wherein the privileged instructions are caused to trap to the virtual machine monitor by using a kernel module of the operating system to reduce a privilege level of the operating system from a higher privilege level.

9. (Previously Presented) The method of claim 6, wherein interposing the virtual machine monitor on the CPU further includes disabling physical memory access by the operating system.

10. (Currently Amended) The method of claim 6, wherein interposing the virtual machine monitor on the CPU further includes loading the virtual machine monitor into the physical memory.

11. (Previously Presented) The method of claim 10, further comprising using a kernel module of the operating system to allocate memory within the operating system, pin the allocated memory, and load the virtual machine monitor into the pinned memory.

12. (Currently Amended) The method of claim 5, ~~wherein the computer hardware includes memory; and~~ wherein the virtual machine monitor is also interposed on the physical memory.

13. (Currently Amended) The method of claim 12, wherein interposing the virtual machine monitor on the physical memory includes partitioning the physical memory to provide partitions, and giving the virtual machine monitor access to at least one of the partitions.

14. (Currently Amended) The method of claim 12, wherein interposing the virtual machine monitor on the physical memory includes using a kernel module of the operating system to allocate a block of the physical memory, pin the block to prevent the operating system from using the block, and allocate the pinned block to the virtual machine monitor.

1 15. (Currently Amended) The method of claim 12, wherein interposing the virtual machine  
2 monitor on the physical memory includes commencing using the virtual machine monitor at  
3 runtime to manage memory translation.

1 16. (Currently Amended) The method of claim 5, ~~wherein the computer hardware includes an~~  
2 ~~I/O device, and~~ wherein the virtual machine monitor is also interposed on the I/O device.

1 17. (Previously Presented) The method of claim 16, wherein the operating system includes a  
2 dual-mode driver that performs direct hardware control in a first mode and communicates with a  
3 device driver of the virtual machine monitor in a second mode; and wherein interposing the  
4 virtual machine monitor on the I/O device includes:  
5 setting the dual-mode driver to the second mode; and  
6 redirecting I/O interrupts to interrupt handlers in the virtual machine monitor instead of to  
7 interrupt handlers in the operating system.

1 18. (Previously Presented) The method of claim 16, wherein interposing the virtual machine  
2 monitor on the I/O device includes commencing I/O emulation of the I/O device at runtime.

1 19.-20. (Cancelled)

1 21. (Currently Amended) The method of claim ~~[[20]]~~, wherein ~~the virtualized computer~~  
2 ~~hardware further includes physical memory, and~~ the virtual machine monitor and the operating  
3 system each include CPU interrupt handlers; and wherein devirtualizing the ~~[[CPU]]~~ at least a  
4 portion of the computer hardware includes redirecting interrupts to the corresponding operating  
5 system CPU interrupt handlers instead of to the virtual machine monitor CPU interrupt handlers.

1 22. (Currently Amended) The method of claim 21, wherein devirtualizing the ~~[[CPU]]~~ at least  
2 a portion of the computer hardware further includes restoring a privilege level of the operating  
3 system from a less privileged mode to a more privileged mode.

23. (Currently Amended) The method of claim 21, wherein devirtualizing the ~~[[CPU]]~~at least a portion of the computer hardware further includes enabling ~~physical memory access of the~~physical memory by the operating system.

24. (Currently Amended) The method of claim 21, wherein devirtualizing the ~~[[CPU]]~~at least a portion of the computer hardware further includes unloading the virtual machine monitor from the physical memory.

25. (Currently Amended) The method of claim ~~[[19]]~~1, ~~wherein the virtualized computer hardware includes memory, and~~ wherein the physical memory is devirtualized at runtime.

26.-27. (Cancelled)

28. (Currently Amended) The method of claim ~~[[19]]~~1, ~~wherein the virtualized computer hardware includes an I/O device, and~~ wherein the I/O device is devirtualized at runtime.

29. (Currently Amended) The method of claim ~~[[28]]~~1, wherein the operating system includes a dual-mode driver that performs direct hardware control in a first mode and communicates with a device driver of the virtual machine monitor in a second mode; and wherein devirtualizing the ~~I/O device~~at least a portion of the computer hardware includes:  
setting the dual-mode driver to the first mode from the second mode, and  
redirecting I/O interrupts to handlers in the operating system instead of handlers in the virtual machine monitor.

30. (Cancelled)

31. (Currently Amended) A computer comprising hardware, the hardware including a central processing unit (CPU), a physical memory, and an input/output (I/O) device, the physical memory encoded with an operating system, a virtual machine monitor, and code for interposing the virtual machine monitor between the operating system and the hardware at runtime, wherein runtime includes a period of execution in the computer after booting and before shutdown of the computer, and wherein the interposing occurs after booting of the computer,

wherein the operating system is to be booted ~~[[in]]~~on the hardware such that the operating system has direct control of at least a portion of the hardware~~computer~~ before interposing the virtual machine monitor, wherein interposing the virtual machine monitor gives the virtual machine monitor direct control of the at least a portion of the hardware such that the operating system no longer has direct control of the at least a portion of the hardware, and

wherein the physical memory is further encoded with code for devirtualizing the at least a portion of the hardware at runtime after the virtual machine monitor has been interposed at runtime, wherein devirtualizing the at least a portion of the hardware comprises stopping the virtual machine monitor such that the operating system resumes having direct control of the at least a portion of the hardware.

32. (Currently Amended) The computer of claim 31, ~~wherein the hardware further includes a CPU~~, wherein the virtual machine monitor is interposed on the CPU at runtime, and the virtual machine monitor and the operating system each include CPU interrupt handlers; and wherein the interposing code is to cause privileged instructions to trap to the virtual machine monitor, and to redirect interrupts and traps to the corresponding virtual machine monitor CPU interrupt handlers instead of to the operating system CPU interrupt handlers.

33. (Previously Presented) The computer of claim 32, wherein the interposing code is to cause privileged instructions to trap to the virtual machine monitor by causing the operating system to run at a reduced privilege level from a higher privilege level; and wherein the interposing code is to reduce a privilege level of the operating system after redirecting the interrupts, and to return control to the operating system at the reduced privilege level.

34. (Previously Presented) The computer of claim 32, wherein the interposing code includes a kernel module of the operating system for reducing a privilege level of the operating system from a higher privilege level, whereby the privileged instructions trap to the virtual machine monitor.

35. (Currently Amended) The computer of claim 32, wherein the interposing code is to disable ~~physical memory access~~ to the physical memory by the operating system.

36. (Currently Amended) The computer of claim 31, wherein the interposing code includes a kernel module of the operating system for allocating a block of the physical memory, pinning the block to prevent the operating system from using the block, and allocating the pinned block to the virtual machine monitor, ~~whereby the virtual machine monitor is interposed on the memory at runtime.~~

37. (Currently Amended) The computer claim 31, wherein the interposing code is to commence using the virtual machine monitor at runtime to manage memory translation, ~~whereby the virtual machine monitor is interposed on the memory at runtime.~~

38. (Currently Amended) The computer of claim 31, ~~wherein the hardware further includes an I/O device; and~~ wherein the interposing code includes an operating system dual-mode driver to perform direct hardware control in a first mode and to communicate with a device driver of the virtual machine monitor in a second mode; and wherein the interposing code is to set the dual-mode driver to the second mode, and to direct I/O interrupts to interrupt handlers in the virtual machine monitor instead of to interrupt handlers in the operating system, ~~whereby the virtual machine monitor is interposed on the I/O device at runtime.~~

39. (Currently Amended) The computer of claim 31, ~~wherein the hardware further includes an I/O device; and~~ wherein the operating system includes a dual-mode driver to perform direct hardware control in a first mode and to communicate with a device driver of the virtual machine monitor in a second mode; and wherein the interposing code is to set the dual-mode driver to the second mode, and to redirect I/O interrupts to interrupt handlers in the virtual machine monitor instead of to interrupt handlers in the operating system, ~~whereby the virtual machine monitor is interposed on the I/O device.~~

40. (Currently Amended) The computer of claim 31, ~~wherein the hardware further includes an I/O device; and~~ wherein the interposing code is to commence I/O emulation of the I/O device at runtime, ~~whereby the virtual machine monitor is interposed on the I/O device at runtime.~~

41.-42. (Cancelled)

43. (Currently Amended) The computer of claim ~~[[42]]~~31, wherein the ~~memory is further encoded with an~~ operating system including includes interrupt handlers; wherein the virtual machine monitor includes interrupt handlers; and wherein the devirtualizing code is to redirect interrupts to the corresponding interrupt handlers of the operating system instead of to the interrupt handlers of the virtual machine monitor.

44. (Currently Amended) The computer of claim 43, wherein the devirtualizing code is to restore a privilege level of the operating system from a lower privilege level to a higher privilege level.

45. (Currently Amended) The computer of claim 43, wherein the devirtualizing code is to enable ~~physical memory access~~ of the physical memory by the operating system.

46. (Currently Amended) The computer of claim ~~[[41]]~~31, wherein the devirtualizing code is to devirtualize the physical memory at runtime.



47. (Previously Presented) The computer of claim 46, wherein the virtual machine monitor is to allocate memory from an operating system to the virtual machine monitor; and wherein the devirtualizing code is to return the allocated memory to the operating system.

48. (Cancelled)

49. (Currently Amended) The computer of claim ~~[[41]]~~31, ~~wherein the hardware includes an I/O device, wherein the virtual machine monitor is to virtualize the I/O device; and wherein the~~ devirtualizing code is to devirtualize the I/O device at runtime.

50. (Currently Amended) The computer of claim 49, wherein the ~~memory is further encoded with an operating system including~~includes dual-mode drivers to perform direct hardware control in a first mode and communicate with device drivers of the virtual machine monitor in a second mode; and wherein the devirtualizing code is to set the dual-mode drivers to the first mode from the second mode, and to redirect I/O interrupts to handlers in the operating system instead of to handlers in the virtual machine monitor.

51. (Previously Presented) The computer of claim 49, wherein the devirtualizing code is to cease emulation of the I/O device at runtime.

52. (Currently Amended) An article for use with an operating system on computer hardware, the article comprising a computer-readable storage medium storing software that when executed by ~~[[the]]~~ a computer causes the computer to:

boot the operating system on the computer hardware such that the operating system has direct control of at least a portion of the computer hardware including a central processing unit (CPU), a physical memory, and an input/output (I/O) device;

~~virtualize at least a portion of the computer hardware at runtime by providing interpose a virtual machine monitor between the operating system and the computer hardware at runtime, wherein runtime includes a period of execution in the computer after booting and before shutdown of the computer, wherein the virtualizing interposing occurs after booting of the computer, and wherein interposing the virtual machine monitor gives the virtual machine monitor direct control of the at least a portion of the computer hardware such that the operating system no longer has direct control of the at least a portion of the computer hardware; and~~

devirtualize the at least a portion of the computer hardware at runtime after the virtual machine monitor has been interposed at runtime, wherein devirtualizing the at least a portion of the computer hardware comprises stopping the virtual machine monitor such that the operating system resumes having direct control of the at least a portion of the computer hardware,

and loading of the operating system, and

~~wherein the operating system is to be booted in the computer before virtualizing the at least a portion of the computer hardware at runtime.~~

53. (Currently Amended) The article of claim 52, ~~wherein the computer hardware further includes a CPU, and~~ wherein the virtual machine monitor and the operating system each include CPU interrupt handlers; and wherein the software is executable to cause privileged instructions to trap to the virtual machine monitor, and to cause interrupts and traps to be redirected to the corresponding virtual machine monitor interrupt handlers instead of to the operating system interrupt handlers.

54. (Previously Presented) The article of claim 53, wherein the software is executable to cause the privileged instructions to trap to the virtual machine monitor by reducing a privilege level of the operating system from a higher privilege level, and wherein the software causes control to be returned to the operating system at the reduced privilege level.

55. (Currently Amended) The article of claim 53, wherein the software is executable to cause ~~physical memory access by the physical memory~~ by the operating system to be disabled.

56. (Currently Amended) The article of claim 52, ~~wherein the computer hardware includes memory, and~~ wherein the virtual machine monitor is for causing a kernel module of the operating system to allocate a block of the physical memory, pin the block to prevent the operating system from using the block, and allocate the pinned block to the virtual machine monitor.

57. (Cancelled)

58. (Currently Amended) The article of claim 52, ~~wherein the computer hardware further includes an I/O device; and~~ wherein the software includes an operating system dual-mode driver to perform direct hardware control in a first mode and communicate with a corresponding device driver of a virtual machine monitor in a second mode; and wherein the dual-mode driver is set to the second mode when the at least the portion of the computer hardware is virtualized, and wherein I/O interrupts are redirected to interrupt handlers in the virtual machine monitor instead of interrupt handlers in the operating system.

59. (Currently Amended) The article of claim 52, ~~wherein the computer hardware further includes an I/O device; and~~ wherein the operating system includes a dual-mode driver to perform direct hardware control in a first mode and communicate with a device driver of the virtual machine monitor in a second mode; and wherein the dual-mode driver is set to the second mode when the at least the portion of the computer hardware is virtualized, and wherein I/O interrupts are redirected from interrupt handlers in the operating system to interrupt handlers in the virtual machine monitor.

60. (Currently Amended) The article of claim 52, ~~wherein the computer hardware further includes an I/O device; and~~ wherein the software is executable to cause I/O emulation of the I/O device to commence at runtime.

61.-62. (Cancelled)

63. (Currently Amended) The article of claim ~~[[62]]~~52, ~~wherein the virtualized hardware includes a CPU; and~~ wherein the software causes the CPU to be devirtualized at runtime.

64. (Currently Amended) The article of claim 63, ~~wherein the virtualized hardware further includes memory; and wherein the memory is further encoded with the operating system including~~ ~~includes~~ first interrupt handlers; wherein the software includes second interrupt handlers; and wherein the software is executable to ~~cause~~devirtualize by causing interrupts to be redirected to the corresponding first interrupt handlers instead of to the second interrupt handlers.

65. (Currently Amended) The article of claim 64, wherein the software is executable to ~~cause~~devirtualize by causing a privilege level of the operating system to be restored from a lower privilege level to a higher privilege level.

66. (Currently Amended) The article of claim 64, wherein the software is executable to devirtualize by causing ~~cause physical memory access of the physical memory~~ by the operating system to be enabled.

67. (Currently Amended) The article of claim ~~[[62]]~~52, ~~wherein the virtualized hardware includes a memory; and~~ wherein the software is executable to cause the physical memory to be devirtualized at runtime.

68. (Currently Amended) The article of claim 67, wherein if a part of the physical memory was allocated from ~~[[an]]the~~ operating system to the virtual machine monitor prior to the runtime devirtualization, the software is executable to cause the allocated physical memory to be returned to the operating system as part of the runtime devirtualization.

69. (Currently Amended) The article of claim 67, wherein the software is executable to cause the physical memory to be remapped and wherein the software allows ~~[[an]]the~~ operating system to manage address translation with respect to the devirtualized physical memory.

70. (Currently Amended) The article of claim ~~[[62]]52~~, ~~wherein the virtualized hardware includes an I/O device; and~~ wherein the software is executable to cause the I/O device to be devirtualized at runtime.

71. (Currently Amended) The article of claim 70, ~~wherein the virtualized hardware further includes a memory; and wherein the memory is further encoded with the operating system including~~includes dual-mode drivers that perform direct hardware control in a first mode and communicate with virtual device drivers in a second mode; and wherein the software is executable to ~~cause~~devirtualize by causing the dual-mode drivers to be set to the first mode.

72. (Currently Amended) The article of claim 70, wherein the software is executable to ~~cause~~devirtualize by causing emulation of the I/O device to cease at runtime.

73.-74. (Cancelled)